IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: M. Barturen et al.   Examiner: Chrystine Pham
Serial No.: 09/943,563   Group Art Unit: 2192
Filed: August 30, 2001   Confirmation No. 2529
For: INTEGRATED SYSTEM AND METHOD FOR THE MANAGEMENT OF A
        COMPLETE END-TO-END SOFTWARE DELIVERY PROCESS

                                                     Date: August 30, 2006

APPELLANTS' BRIEF ON APPEAL UNDER 37 C.F.R. § 41.37

This Appeal Brief is filed pursuant to the *Notice of Appeal to the Board of Patent Appeals and Interferences* filed August 23, 2006. It is not believed that an extension of time and/or additional fee(s) are due. If any additional fee or extension of time is required, Appellants request that this be considered a petition therefor. The Commissioner is authorized to charge any additional fee which may be required, or credit any refund, to our Deposit Account No. 09-0457.

Real Party In Interest

The real party in interest is assignee International Business Machines Corporation, Armonk, New York.

Related Appeals and Interferences

Appellant is aware of no interferences or appeals that would be affected by the present appeal. The present case was previously appealed to the Board of Appeals & Interferences on October 21, 2005. In response to Appellants' Appeal Brief, the Examiner withdrew the pending rejections, but proceeded to reject all of the claims on new grounds. The present appeal challenges these new grounds of rejection.

Status of Claims

Claims 1-17 remain pending, each of which is finally rejected. Appellants appeal the final rejection of Claims 1-17. The attached Claims Appendix presents the pending claims as finally rejected in the Final Office Action ("Final Action") of May 23, 2006 and the Advisory Action of August 11, 2006.

Status of Amendments

The attached Claims Appendix presents the claims as they currently stand. An *Amendment* was filed in this case on March 8, 2005 in which Claim 6 was amended and new Claims 12-17 were added. This *Amendment* was entered. A *Request for Reconsideration* (that did not include any claim amendments) was filed on August 11, 2005 in response to the Final Office Action of June 16, 2005. No *Advisory Action* was received in response to the *Request for Reconsideration*. Appellants' filed a *Notice of Appeal* and then a first *Appeal Brief*. In lieu of responding to the *Appeal Brief*, the Examiner issued an *Office Action* on January 30, 2006 withdrawing the previous rejections, but rejecting all of the pending claims on new grounds. Appellants filed a *Request for Reconsideration* on March 2, 2006 (that did not include any claim amendments). Thereafter, a *Final Office Action* was issued on May 23, 2006. Appellants filed an *Amendment After Final* on June 14, 2006, which amended various of the claims. However, the August 11, 2006 *Advisory Action* refused to enter these claim amendments on the basis that they raised new issues. Accordingly, the claims of the present appeal are the claims presented in the March 8, 2005 *Amendment*.

Summary of Claimed Subject Matter

I.   Independent Claim 1

Independent Claim 1 is directed to an integrated data processing system that manages the delivery of software products to target computers or other target processing units. The language of Claim 1 is repeated below with reference numerals included that indicate the components shown in the embodiment of Figs. 1 and 2 of the present application that correspond to the components recited in Claim 1:

> 1.   An integrated data processing system (201) for managing a process of delivery of software products to target software product execution units (103) in a network environment, comprising:
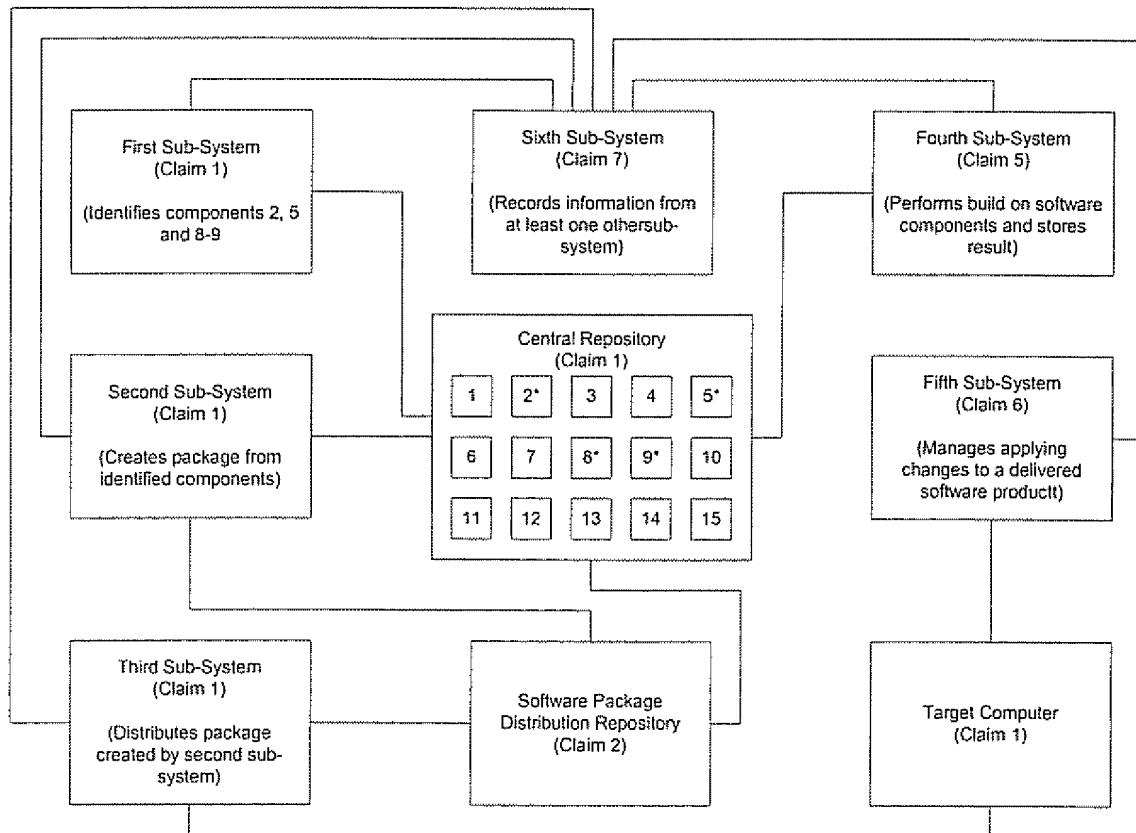>
> a central repository (215) for storing software components of at least one software product;
>
> a first sub-system (203) for identifying within the central repository software components of a software product to be delivered;

a second sub-system (205) for creating at least one software product package from the identified software components identified by the first sub-system, and

a third sub-system (209) for distributing the at least one software product package created by the second sub-system (205) to the target software product execution units (103).

To further illustrate the invention of Claim 1 (and the inventions of dependent Claims 2-7 which depend from Claim 1), Appellants provide below a block diagram (which is a combined and expanded version of FIGS. 1 and 2 of the present application) of an integrated data processing system according to various embodiments of the invention of Claims 1-7. In the block diagram, all of the subsystems recited in any of Claims 1-7 are included, and connectors are provided that show how the various subsystems can be interrelated in one specific embodiment of the present invention. It will be understood that other embodiments of the invention may have fewer elements and different connections.

As shown in the diagram, the integrated data processing system according to Claim 1 includes a Central Repository that stores the software components required for at least one software product. These components are illustrated in the block diagram by the small numbered boxes included within the Central Repository. The integrated data processing system according to Claim 1 further includes a First Sub-System that is used to identify the software components within the Central Repository that are needed to build and/or deliver a target software product to a target end-user computer (or other execution unit). In the exemplary block diagram, the First Sub-System has identified software components 2, 5, 8 and 9 (as indicated by the "*" after each of these components in the Central Repository). The integrated data processing system according to Claim 1 further includes a Second Sub-System that creates one or more software product packages using the software components identified by the First Sub-System. In certain embodiments of the present invention (see Claim 2), the software product package that is created by the Second Sub-System may be stored in a separate Software Package Distribution Repository (in other embodiments it may be stored in other locations such as, for example, the Central Repository). Finally, the integrated data processing system according to Claim 1 includes a Third Sub-System that distributes the software product package created by the Second Sub-System to the target end-user computer.

As is also shown in the block diagram above and discussed in Claim 5, in certain embodiments of the present invention, a Fourth Sub-System may be provided that performs a building process using at least some of the identified software components. The resulting components generated by the build process may then be stored, for example, in the Central Repository. In addition, in certain embodiments (see Claim 6), a Fifth Sub-System may be provided that manages the process of applying changes to software products that have already been delivered. In still other embodiments (see Claim 7), a Sixth Sub-System may be provided that records information provided by one or more of the other sub-systems.

II. Independent Claim 8

Independent Claim 8 is directed to a method for delivering software products to target software product execution units (boxes 103 in Fig. 1). Pursuant to the methods of Claim 8,

software components of one (but often more than one) software product(s) are stored in a central repository (box 215 in Fig. 2). The software components in the central repository (215) that correspond to a specific software product that is to be delivered are then identified. Next, a software product package is created that includes at least one of the identified software components. This software product package is then distributed to, and installed on, the target software product execution units (103).

III. Independent Claim 12

Independent Claim 12 is directed to methods of developing and installing a software product on a group of target computers (boxes 103 in Fig. 1). Pursuant to the methods of Claim 12, a plurality of components are stored in a central repository (box 215 of Fig. 2). At least some of the stored components are used to build the software product. Once built, the software product is returned to the central repository (215). An installable software package is then created, where the installable software package includes at least some of the plurality of components and the built software product. This installable software package is stored in a second repository. The installable software package is then distributed to, and installed on, at least some of the target computers (103).

Grounds of Rejection to be Reviewed on Appeal

1.   The rejections of Claims 1-2, 4-8, 10, 12-15 and 17 under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 6,427,230 to Goiffon et al. ("Goiffon").

2.   The rejections of Claims 3 and 9 under 35 U.S.C. § 103(a) as obvious over Goiffon in view of U.S. Patent No. 5,974,454 to Apfel et al. ("Apfel").

3.   The rejection of Claims 11 and 16 under 35 U.S.C. § 103(a) as being unpatentable over Goiffon in view of U.S. Patent No. 6,110,228 to Albright et al. ("Albright").
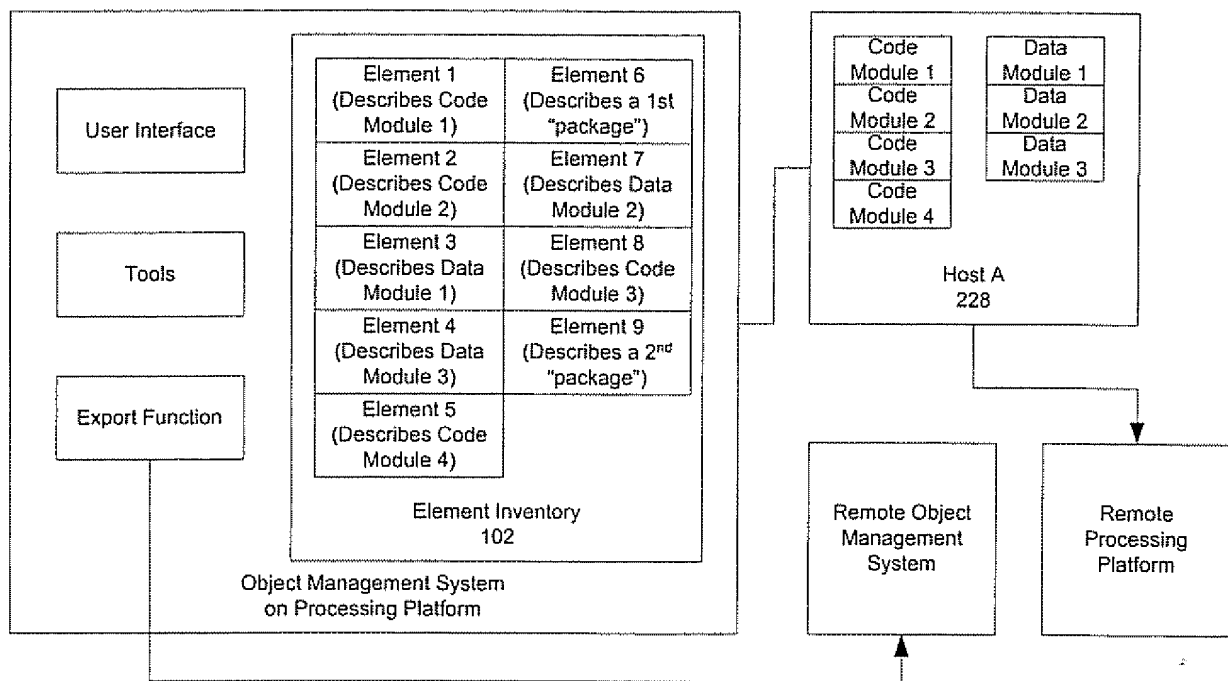
Argument

I.   The System of Goiffon

As noted above, all but four of the pending claims stand rejected as anticipated by Goiffon. Before addressing the pending rejections, it is helpful to compare and contrast the

system disclosed in Goiffon to the systems and methods of the present invention which are briefly described in the *Summary of Claimed Subject Matter* section above. Goiffon is directed to an Object Management System that is used to catalog and manage code and data modules that may be re-used pursuant to object-oriented programming techniques. (*See, e.g.,* Goiffon at Col. 1, lines 26-31; Col. 2, lines 50-52). In particular, in the system of Goiffon, "objects" are created that are associated with respective ones of the potentially re-useable code or data module.[1] (Goiffon at Col. 4, lines 3-6). Each object/element contains data about its corresponding one of the code or data modules such as, for example, the location of the code or data module, its type and, most importantly, its relationship and interdependencies with respect to others of the code or data modules. (*See* Goiffon at Col. 6, line 62 through Col. 7, line 5). The objects/elements thus comprise "meta-data", which Goiffon explains is "data about data." (Goiffon at Col. 6, line 62).

The system of Goiffon provides a user interface that allows a user to select a group of code or data modules that they want to reuse in another application. (Goiffon at Col. 2, lines 24-38, 53-61; Col. 4, lines 15-21). The system then identifies all of the other code and data modules that will be needed (e.g., other code or data modules that are called by the selected code/data modules). (Goiffon at Col. 2, lines 24-38, 53-61; Col. 4, lines 15-21 and lines 29-48). In this manner, the system helps a user to ensure that all of the necessary code or data modules are included in the package of code/data modules that is to be reused. (Goiffon at Col. 2, lines 24-38, 53-61; Col. 4, lines 15-21). Appellants have provided below a block diagram which shows the various components of Goiffon that are cited in rejecting the pending claims, and the relationship between those components.

---

[1] In Goiffon, the "objects" that are associated with the code and data modules are also referred to as "elements", and the code and data modules are sometimes generically referred to as "software constructs." (*See, e.g.,* Goiffon at Col. 6, lines 59-60; Col. 2, lines 1-2).

Object Management System on Processing Platform

- User Interface
- Tools
- Export Function

Element Inventory 102

Element 1 (Describes Code Module 1)	Element 6 (Describes a 1st "package")
Element 2 (Describes Code Module 2)	Element 7 (Describes Data Module 2)
Element 3 (Describes Data Module 1)	Element 8 (Describes Code Module 3)
Element 4 (Describes Data Module 3)	Element 9 (Describes a 2nd "package")
Element 5 (Describes Code Module 4)	

Host A 228

Code Module 1	Data Module 1
Code Module 2	Data Module 2
Code Module 3	Data Module 3
Code Module 4	

Remote Object Management System

Remote Processing Platform

As shown in the above diagram, the system of Goiffon includes an Element Inventory 102 that includes a plurality of "objects" or "elements." (Goiffon at Col. 6, line 56 through Col. 7, line 6). Each object/element in the Event Inventory 102 models or "catalogs" a corresponding code module or data module that is stored elsewhere (in the embodiment of Goiffon they are stored on Host A 228, as shown above). (Goiffon at Col. 6, line 56 through Col. 7, line 6). The meta-data contained in each object/element in the Element Inventory 102 is data that describes, for example, the location of the corresponding code or data module, its type and, most importantly, its relationship and interdependencies with respect to others of the code or data modules. (Goiffon at Col. 6, line 56 through Col. 7, line 6). Element Inventory 102 may also include objects/elements that describe the interdependencies amongst various groupings of code or data modules, which are also referred to in Goiffon as "packages." (*See* Goiffon at Col. 2, lines 45-47 and Col. 4, lines 59-67). Elements 6 and 9 in the diagram above are examples of such "package objects."

The User Interface of Goiffon allows a user to view the objects stored in Element Inventory 102. The objects show, for example, the relationships that a particular data or code module has with other code and data modules. (Goiffon at Col. 3, lines 1-5 and Col. 4, lines 29-

48). This information helps the user to choose which code or data modules to include in a package of re-useable code and data modules that is to be created for performing some task. (Goiffon at Col. 3, lines 1-5 and Col. 4, lines 29-48). The code and data modules themselves are stored on Host A. (Goiffon at Col. 13, lines 3-6).

The "export function" shown in the diagram above is a feature that allows a user to export some or all of the elements (i.e., meta-data objects that describe software) to another, remote, Object Management System. (*See* Goiffon at Col. 14, lines 21-25). Groups of code and data modules ("packages") may also be "migrated" to another processing platform as shown in the above diagram. (*See* Goiffon at Col. 4, lines 21-26).

II.   Legal Standards

A.   Legal Standards Under 35 U.S.C. § 102

Most of the pending claims stand rejected as anticipated under 35 U.S.C. § 102. Under 35 U.S.C. § 102, "a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." M.P.E.P. § 2131; *Verdegaal Bros. v. Union Oil Co.*, 814 F.2d 628, 631 (Fed. Cir. 1987). Anticipation requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention. *Apple Computer Inc. v. Articulate Sys. Inc.*, 57 U.S.P.Q.2d 1057, 1061 (Fed. Cir. 2000).

A finding of anticipation further requires that there must be no difference between the claimed invention and the disclosure of the cited reference as viewed by one of ordinary skill in the art. *See Scripps Clinic & Research Foundation v. Genentech Inc.*, 927 F.2d 1565, 1576, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991). In particular, the Court of Appeals for the Federal Circuit held that a finding of anticipation requires absolute identity for each and every element set forth in the claimed invention. *See Trintec Indus. Inc. v. Top-U.S.A. Corp.*, 63 U.S.P.Q.2d 1597 (Fed. Cir. 2002). Additionally, the cited prior art reference must be enabling, thereby placing the allegedly disclosed matter in the possession of the public. *In re Brown*, 329 F.2d 1006, 1011, 141 U.S.P.Q. 245, 249 (C.C.P.A. 1964). Thus, the prior art reference must adequately describe the claimed invention so that a person of ordinary skill in the art could make and use the invention.

B.   Legal Standards Under 35 U.S.C. § 103

The remaining claims are rejected as obvious under 35 U.S.C. § 103. To establish a *prima facie* case of obviousness, the prior art reference or references when combined must teach or suggest *all* the recitations of the claims, and there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. M.P.E.P. § 2143. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. M.P.E.P. §2143.01, citing *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990). As emphasized by the Court of Appeals for the Federal Circuit, to support combining references, evidence of a suggestion, teaching, or motivation to combine must be clear and particular, and this requirement for clear and particular evidence is not met by broad and conclusory statements about the teachings of references. *In re Dembiczak*, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). The Court of Appeals for the Federal Circuit has further stated that, to support combining or modifying references, there must be particular evidence from the prior art as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed. *In re Kotzab*, 55 U.S.P.Q.2d 1313, 1317 (Fed. Cir. 2000).

III.   The Rejections of Claims 1-2, 4-8, 10, 12-15 and 17 as Anticipated by Goiffon Should be Reversed

Claims 1-2, 4-8, 10, 12-15 and 17 stand finally rejected as anticipated under 35 U.S.C. § 102(b) by Goiffon. (Final Office Action at 5-6 and 8). For the reasons discussed below, Appellants respectfully submit that the rejection of each of these claims should be reversed.

A.   Claims 1, 4-6 and 8 are Patentable Over Goiffon

Independent Claim 1 is directed to an integrated data processing system. Independent Claim 8 is a method claim that corresponds to Claim 1, and includes very similar recitations. Accordingly, while Appellants' arguments will be presented herein with respect to Claim 1, it will be appreciated that the same arguments apply to Claim 8. Claims 4-6 depend from Claim 1, and

hence separate arguments against the rejections of Claims 4-6 will not be presented herein, as each of these claims are patentable over Goiffon for at least the reasons that Claim 1 is patentable over Goiffon. Independent Claim 1 recites:

> 1.   An integrated data processing system for managing a process of delivery of software products to target software product execution units in a network environment, comprising:
>
> a central repository for storing software components of at least one software product;
>
> a first sub-system for identifying within the central repository software components of a software product to be delivered;
>
> a second sub-system for creating at least one software product package from the identified software components identified by the first sub-system, and
>
> a third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units.

The Final Action points to numerous different components of Goiffon as allegedly disclosing each of the above recitations of Claim 1. The basis for contending that these disparate components allegedly disclose the invention of Claim 1 is far from clear. By way of example, the Final Action states that the "central repository for storing software components of at least one software product" recited in the first clause of the body of Claim 1 is taught by Goiffon's disclosure of (a) an object repository, (b) software constructs, (c) packages, (d) AIM Server 214, (e) Element Repository 220, (f) Host A 228, (g) Memory 229, and (h) data modules. (*See* Final Action at 5, listing these components and citing to FIG. 2B and Col. 12, lines 7-15, 23-67 and Col 12, line 57 through Col. 13, line 20 of Goiffon). The Final Action, however, does not even attempt to explain what combination of the eight (8) separate items enumerated above allegedly constitutes the "central repository for storing software components of at least one software product" of Claim 1. The Final Action similarly cites to a laundry list of components from Goiffon as disclosing each of the first through third sub-systems recited Claim 1. (*See* Final Action at 5-6). For at least the reasons discussed in the following sections, Appellants respectfully submit that Goiffon does not disclose or suggest the system of Claim 1.

1.   The Cited Portions of Goiffon Do Not Disclose a System for
Managing a Process of Delivery of Software Products

Claim 1 is directed to, among other things, a "system for managing a process of delivery

of software products." The Final Action states that Goiffon, at the Abstract, Col. 7, lines 23-40,

Col. 14, lines 20-25 and Figs. 2A and 2B, discloses such a "system for managing a process of

delivery of software products." Appellants respectfully submit, however, that the cited portions

of Goiffon do not disclose such a system. In particular, Col. 7, lines 23-40 and Col. 14, lines 20-

25 of Goiffon describe the "**Element** Inventory 102" and an "export" operation that may be used

to "provide a copy of an **element** to the remote system." (*See, e.g.,* Goiffon at Col. 7, lines 33-

34). Goiffon clearly and repeatedly states that the **elements** that are stored in the Element

Inventory 102 and that are exported pursuant to the export operation are **objects that store meta**

data describing the location, type and various other attributes of data and code modules that are

stored elsewhere in the system. Goiffon states, for example:

> **Element Inventory 102 . . . stores the various objects, or "elements", that are**
> **used to manage the code and data components** (not shown in FIG. 1) that support
> an enterprise. Each of the **objects stores meta-data, or "data about data"**. This
> meta-data describes, among other things, the location of, and the type of, data or code
> that is stored within the respective component or module residing elsewhere within
> the system. This meta-data stored in an element also describes the various
> relationships that the respective data or code module has with other data and/or code
> modules. In this manner, **the elements stored in the Element Inventory 102 serves**
> **as an index that points to, and describes, the various data and code resources**
> **used to perform the functions** . . . .

(Goiffon at Col. 6, line 58 through Col. 7, line 6) (emphasis added). Similarly, at Col. 10, lines

16-19, Goiffon states:

> [E]ach of these elements includes meta-data that describes the location and function of
> the associated code or data element. This meta-data will further describe the
> relationships that an element has with other elements . . . .

Similar descriptions of the objects stored in Element Inventory are provided throughout Goiffon.

(*See, e.g.,* Goiffon at Col. 2, lines 57-61, Col. 4, lines 3-8; Col. 11, lines 5-11). Goiffon also

repeatedly states that the actual code and data modules are **not** stored in the Element Inventory

102, but are instead stored elsewhere in the system. (*See, e.g.,* Goiffon at Col. 6, lines 59-66,

stating that meta-data objects are stored in the Element Inventory 102 which "describe, among other things, the location of, and the type of, data or code . . . **residing elsewhere within the system**"; *see also* Goiffon at Col. 11, lines 35-38, stating that "the code and data modules are **stored elsewhere**") (emphasis added). In fact, Goiffon expressly states that the actual code and data modules are stored in Host A 228, which is a separate element that is interconnected to the Data Processing System 229 that runs the Object Management System by a network interconnection. (*See* Goiffon at Col. 13, lines 3-6 and Fig. 2B).

As is clear from the above-cited excerpts of Goiffon, the **elements** stored in Element Inventory 102 are meta-data objects that include **information** (i.e., data) about respective ones of the actual code and data modules that are used to create a software package. This information may include the location of the code or data modules, the type of module, its interdependencies with other modules (i.e., what subroutine modules it calls), etc. As such, the "export function" described at Col. 14, lines 21-25 of Goiffon clearly does **not** involve the delivery of a software product to an execution unit, but instead involves the export of a meta-data object that points to, and describes, a software object that may be used to perform a function. Likewise, the "object repository" of Goiffon describes the location where the meta-data objects are stored, and the "elements" of Goiffon are the actual meta-data objects.

In the Response to Arguments section of the Final Action, the Examiner cites to the Objects and Summary of Invention sections of Goiffon to argue that the "element packages" of Goiffon "are actual software constructs, and not merely 'metadata' as asserted by Applicants." (Final Action at 2-3). However, the portions of Goiffon cited by the Examiner describe software packages that are formed by grouping a plurality of code and data modules. Appellants do not dispute that Goiffon discloses that codes and data modules may be grouped together to form a package, and that such a package of actual software modules can be "migrated" to a new platform. (*See, e.g.,* Goiffon at Col. 3, lines 27-32 and Col. 4, lines 15-26). However, these "packages" indisputably are not stored in the Element Inventory 102, and have nothing to do with the element "export function" of Goiffon that forms the basis for the pending rejection of Claim 1. Appellants respectfully submit that the Examiner does not attempt to argue that the description in Goiffon of the "migration" of packages of software constructs (i.e., the actual code

and data modules) anticipates the pending claims because that description does not disclose numerous recitations of the claims. The Examiner cannot overcome this by improperly mixing and matching descriptions of what is done with the actual code and data modules and the meta-data objects as has been done in the pending rejections. Accordingly, the rejection of Claim 1 should be reversed, as the cited portions of Goiffon do not teach or disclose a "system for managing a process of delivery of software products."

> 2. The Cited Portions of Goiffon Do Not Disclose a "Central Repository for Storing Software Components of at Least One Software Product"

Appellants also respectfully submit that the cited portions of Goiffon do not disclose a "central repository for storing software components of at least one software product" as recited in Claim 1. While neither the Final Action nor the Advisory Action is particularly clear, it appears that the Examiner has taken the position that the Element Inventory 102 that is part of Element Repository 220 of Fig. 2B of Goiffon comprises the "central repository" of Claim 1, and that the data and code modules of Goiffon (*see, e.g.*, Goiffon at Col. 3, lines 16-17) comprises the "plurality of components" that are stored in the central repository. (Final Action at p. 5).

As discussed in detail above, the Element Inventory 102 of Goiffon does **not** comprise "a central repository for storing software components of at least one software product" that is ultimately delivered to target execution units as recited in the first recitation of the body of Claim 1. Instead, Goiffon teaches that:

> **Element Inventory 102 . . . stores the various objects, or "elements", that are used to manage the code and data components** (not shown in FIG. 1) . . . . Each of the **objects stores meta-data**, or "data about data". This meta-data describes, among other things, the location of, and the type of, data or code that is stored within the respective component or module residing elsewhere within the system.

(Goiffon at Col. 6, lines 58-66) (emphasis added). Thus, Goiffon very clearly states the Element Inventory is used to store objects that contain **data** about the actual software building blocks as opposed to the code and data modules themselves. While Appellants do not dispute that Goiffon discusses creating packages of a plurality of code and data modules that may be used to perform a given task (*see* Goiffon at Col. 4, lines 15-26), such code and data modules are **not** what is

stored in the Element Inventory/Repository. Instead, the above-quoted excerpt of Goiffon expressly states that (1) the Element Inventory/Repository stores meta data objects/elements and (2) that the code and data modules "resid[e] elsewhere within the system." Consequently, the cited portions of Goiffon do not disclose the first recitation of the body of Claim 1, and hence the rejection of Claim 1 should be reversed for this additional reason.

3. The Cited Portions of Goiffon Do Not Disclose a "Third Sub-System for Distributing the at Least one Software Product"

Appellants also respectfully submit that the cited portions of Goiffon do not disclose or suggest the "third sub-system for distributing the at least one software product package" that is recited in Claim 1. The Final Action cites to the terms "export function", "element", "remote system" and "client server" at Col. 2, lines 53-56, Col. 3, lines 20-32, Col. 4, lines 15-67, Col. 7, lines 23-40, Col. 8, lines 47-57, Col. 14, lines 20-25 and Figs. 1, 2A and 2B of Goiffon as disclosing this recitation of Claim 1. (Final Action at 6). However, as noted above, what these portions of Goiffon disclose is that an "export" operation may be used to "provide a copy of an **element** to the remote system." (Goiffon at Col. 7, lines 33-34). As is also discussed above, the "elements" of Goiffon are objects that contain **meta-data** that describes the actual code and data modules that may be used to perform functions. (Goiffon at Col. 6, lines 58-66). Thus, the cited portions of Goiffon only discuss exporting meta data to a "remote system" (which is another Object Management System), and clearly do not teach or disclose "distributing the at least one software product package created by the second sub-system to the target software product execution units" as recited in Claim 1. (*See* Goiffon at Col. 7, lines 29-33). Thus, Appellants respectfully submit that the cited portions of Goiffon do not disclose a sub-system that distributes software products to target execution units, providing another independent basis for reversal of the rejection of Claim 1.

Thus, for at least each of the above reasons, Appellants respectfully submit that Goiffon does not anticipate Claim 1. For the exact same reasons, Appellants submit that Goiffon likewise does not anticipate Claims 4-6 or 8, and therefore respectfully requests reversal of the rejections of each of these claims.

B. Claims 2 and 10 are Patentable Over Goiffon

Claims 2 and 10 likewise stand rejected as anticipated by Goiffon. Claim 2 depends from Claim 1 and Claim 10 depends from Claim 8. Accordingly, the rejections of Claims 2 and 10 should be reversed for the same reasons, discussed above, that the rejections of Claims 1 and 8 should be reversed. In addition, Claims 2 and 10 each recite a "software package distribution repository" which is used to store the created software product package.

The Final Action cites to element 1024 of Fig. 10 and elements 1808, 1816 and 1828 of Figs. 18A and 18B as disclosing the recitations added by Claim 2. However, the cited portions of Goiffon are directed to processing steps that are used to create "Element Packages", which are nothing more than meta-data objects. (*See* Goiffon at Col. 22, lines 31-34, explaining that an "Element Package will comprise elements that represent, and model, all code and data modules that are needed to perform one or more predetermined functions"). As noted above, these meta-data objects are **not** created from an identified group of software components that are stored in a central repository, and hence the creation of such element packages clearly does not disclose the recitations of Claims 2 and 10. Accordingly, Claims 2 and 10 are patentable over the cited art for at least this additional reason.

C. Claim 7 is Patentable Over Goiffon

Claim 7 depends from Claim 1, and thus is patentable for the same reasons, discussed above, that Claim 1 is patentable over Goiffon. In addition, Claim 7 recites that the system also includes "a sixth sub-system for recording information provided by at least one of the first through fifth sub-systems of the integrated data processing system during delivery of the software product."

The Final Action cites to line 227 of Fig. 2A and line 240 of Fig. 2B of Goiffon, and associated text, as disclosing such a subsystem. (*See* Final Action at 8). Line 227 of Fig. 2A of Goiffon is a "File I/O" line that connects Client Server 216 to elements of AIM Server 214. Line 227 is not discussed anywhere in the specification of Goiffon (as indicated by a text search of Goiffon on the term "227"). Appellants respectfully submit that line 227 of Fig. 2A of Goiffon clearly does not disclose or suggest "recording information . . . during delivery of the software

product" as recited in Claim 1. Indeed, there is no indication whatsoever that line 227 of Goiffon "records" anything, let alone does so "during the delivery of a software product." Likewise, line 240 of Fig. 2B of Goiffon shows a connection between the "Import/Export Files" and the "File I/O line 227. Neither line 240 nor the description thereof have anything to do with providing "a sixth sub-system for recording information provided by at least one of the first through third sub-systems of the integrated data processing system during delivery of the software product" as recited in Claim 1. In the Advisory Action, the Examiner argues that the recitation of Claim 7 is inherent in the step of importing a software package from the AIM server to the Client server. (Advisory Action at 2). No support whatsoever is provided for this assertion, and it certainly is not "necessarily the case" that the system of Goiffon "record[s] information provided by at least one of the first through third sub-systems of the integrated data processing system during delivery of the software product" as recited in Claim 7. Accordingly, for each of the above reasons, Appellants respectfully submit that Claim 7 is independently patentable over Goiffon for at least this reason.

D. Claims 12-15 and 17 are Patentable Over Goiffon

Independent Claim 12 stands finally rejected as anticipated under 35 U.S.C. § 102(b) by Goiffon. (Final Office Action at 8-9). Claims 13-15 and 17 depend from Claim 12, and hence are patentable over Goffion for at least the reasons that Claim 12 is patentable over Goiffon. Independent Claim 12 recites:

> 12. A method of developing and installing a software product on a plurality of target computers, the method comprising:
> storing a plurality of components in a central repository;
> using at least some of the plurality of stored components to build the software product;
> storing the built software product in the central repository;
> creating an installable software package that includes at least some of the plurality of components and the built software product;
> storing the installable software package in a second repository;

distributing the installable software package to at least some of the plurality of target computers; and

installing the distributed installable software package on the at least some of the plurality of target computers.

For the reasons discussed below, Appellants respectfully submit that the rejections of Claim 12, and Claims 13-15 and 17 depending therefrom, should be reversed.

1. The Cited Portions of Goiffon Do Not Disclose Recitations of Claim 12 That Are Discussed Above With Respect to Claim 1

As an initial matter, Appellants note that Claim 12 includes several recitations that are similar to the recitations of Claim 1. In particular, the "storing a plurality of components in a central repository" recitation of Claim 12 is similar to the "central repository for storing software components of at least one software product" recitation of Claim 1. Likewise, the "distributing the installable software package to at least some of the plurality of target computers" of Claim 12 is similar to the "distributing the at least one software product package created by the second sub-system to the target software product execution units" recitation of Claim 1. In each case, the Final Action cites to the passages of Goiffon as disclosing the corresponding recitations from Claims 1 and 12. It is respectfully submitted that the same reasons, discussed above, that Goiffon does not teach the above-quoted recitations of Claim 1, the cited portions of Goiffon likewise do not disclose the above-quoted recitations of Claim 12. Thus rather than repeating these grounds for reversal of the rejection of Claim 12, Appellants incorporate the arguments and evidence from the discussion Claim 1 above and request reversal of the rejection of Claim 12 for the same reasons.

2. The Cited Portions of Goiffon Do Not Disclose Storing the Built Software Product in the Central Repository

Appellants also submit that Goiffon does not disclose or suggest "storing the built software product in the central repository." The Final Action cites to block 1024 of Fig. 10 and blocks 1808, 1816 and 1828 of Figs. 18A and 18B as disclosing this recitation of Claim 12. Block 1024 of Fig. 10 and blocks 1808, 1816 and 1828 of Figs. 18A and 18B are directed to

steps in processes that are used to create "Element Packages." (Goiffon at Col. 27, lines 11-13 and Col. 34, lines 60-62). In particular, block 1024 of Fig. 10 of Goiffon states that "an element of type 'elementpackage' [is created] to record the list of elements in the element package." (Goiffon at Fig. 10, block 1024). Blocks 1808, 1816 and 1828 likewise recite steps in a process that are used to create an "Element Package." As discussed above, the "elements" of Goiffon are objects which store meta-data (i.e., data about data) that describes the location, type and relationships of various data or code modules. (*See, e.g.,* Goiffon at Col. 6, line 58 through Col. 7, line 6). Goiffon also **expressly describes the "Element Package" as the "elements that represent, and model, all code and data modules that are needed to perform one or more predetermined functions**." (Goiffon at Col. 22, lines 28-34). Thus, it is clear that the "Element Packages" of Goiffon are not a "built software product", but instead is a package of meta-data objects that describe a set of code and data modules that are located elsewhere. Accordingly, and the rejection of Claim 12 should be reversed for this additional reason.

3.   Goiffon Does Not Disclose Storing Both a Plurality of Components and Software Product Built From Those Components in the Same Central Repository

The combination of the first and third recitations in the body of Claim 12 recite that the built software products be stored in the **same** "central repository" in which the plurality of components are stored. Although it is not entirely clear, it appears that the Examiner is taking the position that the Element Inventory of Goiffon comprises such a central repository. Appellants respectfully submit, however, that there is absolutely no teaching that built software products are stored in the Element Inventory 102 of Goiffon – instead, it is clear that meta-data objects are the only thing stored in the Element Inventory 102. More importantly, there is simply no disclosure in Goiffon of storing both a plurality of components and built software products in the same repository. Thus, the rejection of Claim 12 should also be reversed for this reason.

4.   The Cited Portions of Goiffon Do Not Disclose Storing the Installable Software Product in a Second Repository

Claim 12 further recites "storing the installable software product in a second repository."

The Final Action does not even attempt to identify where this recitation of Claim 12 is disclosed in Goiffon. In the Advisory Action, the Examiner argues that the Inventory Administration function 104 provides the capability to export an "element package" to the Element Inventory 102 of a remote Object Management System, therefore disclosing "storing the installable software product in a second repository" as recited in Claim 12. This argument, however, fails for at least two reasons. First, as noted above, Goiffon repeatedly states that the elements that are stored in the Element Inventory 102 are meta-data objects, as opposed to an "installable software product" as recited in Claim 12. Second, the "installable software package" of Claim 12 is something that is built from the plurality of components stored in the central repository. As noted above, the pending rejections rely on the meta-data objects in the Element Inventory as comprising the "components" of Claim 12. As such, they cannot also comprise an "installable software product" that is built from the plurality of components. Accordingly, the rejection of Claim 12 should be reversed for this additional reason.

IV. The Rejections of Claims 3 and 9 as Obvious Over Goiffon in View of Apfel Should be Reversed

Claims 3 and 9 stand rejected as obvious over Goiffon in view of Apfel. Claim 3 depends from Claim 1, and Claim 9 depends from Claim 8. Accordingly, Claims 3 and 9 are patentable based on the same reasons, discussed above, that Claims 1 and 8 are patentable over the cited art. While Appellants dispute the Examiner's position that Apfel discloses the recitation of Claims 3 and 9, Appellants need not argue this point as Appellants respectfully submit that one of skill in the art would not have been motivated to combine Goiffon and Apfel in the manner suggested in the Office Action. As noted above, Goiffon is directed to a system for managing reusable groups of software. (*See, e.g.,* Goiffon at Title). Apfel, on the other hand, is directed to a method and system for updating software programs that are already resident on target computers. The disclosures of Goiffon and Apfel have nothing in common, and are directed to entirely different problems and solutions. Appellants respectfully submit that a person of skill in the art would not have been motivated to incorporate the disclosure of Apfel regarding different upgrade packages for different operating systems into the system of Goiffon,

which is concerned with tracking interdependencies between individual data and code modules to facilitate building software packages using groups of such modules. Appellants respectfully submit that it is only by using Claims 3 and 9 as a roadmap that one would decide to combine the disclosures of Goiffon and Apfel in the manner suggested in the Final Action. However, as this is not a proper basis for finding motivation to combine references, the rejections of Claims 3 and 9 should also be reversed.

V.   The Rejections of Claims 11 and 16 as Obvious Over Goiffon in View of Albright Should be Reversed

Claims 11 and 16 stand rejected as obvious over Goiffon in view of Albright. Claim 11 depends from Claim 8, and Claim 16 depends from Claim 12. Albright is not cited as disclosing any of the recitations from Claims 8 and 12, which are identified above, that Goiffon fails to disclose. Accordingly, Appellants respectfully submit that Claims 11 and 16 are patentable based on the same reasons, discussed above, that Claims 1 and 8 are patentable over the cited art, and request reversal of the rejections of Claims 11 and 16 for this reason.

VI.   Conclusion

In light of the above discussion, Appellants submit that each of the pending claims is patentable over the cited art and, therefore, request reversal of the rejections of Claims 1-17.
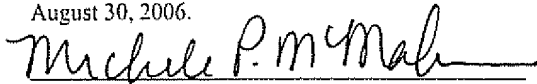
Respectfully submitted,

D. Randal Ayers
Registration No. 40,493

USPTO Customer No. 20792
Myers Bigel Sibley & Sajovec, P.A.
Post Office Box 37428
Raleigh, North Carolina 27627
Telephone: (919) 854-1400
Facsimile: (919) 854-1401

CERTIFICATION OF ELECTRONIC TRANSMISSION
UNDER 37 CFR § 1.8

I hereby certify that this correspondence is being transmitted electronically to the U.S. Patent and Trademark Office on August 30, 2006.

Michele P. McMahan
Date of Signature: August 30, 2006

1. (Original) An integrated data processing system for managing a process of delivery of software products to target software product execution units in a network environment, comprising:

a central repository for storing software components of at least one software product;

a first sub-system for identifying within the central repository software components of a software product to be delivered;

a second sub-system for creating at least one software product package from the identified software components identified by the first sub-system, and

a third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units.

2. (Original) The integrated data processing system according to claim 1, further comprising a software package distribution repository for storing the at least one software product package created by the second sub-system from the identified software components.

3. (Original) The integrated data processing system according to claim 1, in which the third sub-subsystem distributes the at least one software product package to target software product execution units belonging to at least one environment according to at least one role assigned to the at least one software product package by the second sub-system.

4. (Original) The integrated data processing system according to claim 1, in which the first sub-system manages a storage in the central repository of the software components of the software product to be delivered.

5. (Original) The integrated data processing system according to claim 1, further comprising a fourth sub-system for performing a building process of software code components among the identified software components of the software product to be delivered, the fourth sub-system storing a result of the building process in the central repository.

6. (Previously Presented) The integrated data processing system according to claim 1, further comprising a fifth sub-system for managing a process of applying changes to the at least one software product distributed by the third sub-system.

7. (Original) The integrated system according to claim 1, further comprising a sixth sub-system for recording information provided by at least one of the first through fifth sub-systems of the integrated data processing system during delivery of the software product.

8. (Original) A method for delivering software products to target software product execution units in a network environment, comprising the steps of:

storing software components of at least one software product in a central repository;

identifying software components of a software product to be delivered among the software components stored in the central repository;

creating at least one software product package that includes at least one of the identified software components;

distributing the software product package to the target software product execution units and installing the software product package thereon.

9. (Original) The method according to claim 8, in which the step of creating at least one software product package includes assigning to the at least one software product package an indication of role for identifying the target software product execution units to which the software product is to be distributed, and distributing the at least one software product package according to the indication of role.

10. (Original) The method according to claim 8, further comprising a step of storing the at least one software product package in a software distribution repository.

11. (Original) The method according to claim 10, further comprising a step of building identified source code components of the software product to be delivered stored in the central repository, and storing the result of the building in the central repository.

12. (Previously Presented) A method of developing and installing a software product on a plurality of target computers, the method comprising:

storing a plurality of components in a central repository;

using at least some of the plurality of stored components to build the software product;

storing the built software product in the central repository;

creating an installable software package that includes at least some of the plurality of components and the built software product;

storing the installable software package in a second repository;

distributing the installable software package to at least some of the plurality of target computers; and

installing the distributed installable software package on the at least some of the plurality of target computers.

13. (Previously Presented) The method of Claim 12, wherein the software product comprises a newly developed software product.

14. (Previously Presented) The method of Claim 12, wherein the software product comprises a new release and/or a new version of an already released software product.

15. (Previously Presented) The method of Claim 12, further comprising recording information regarding the software product in a tracking sub-system.

16. (Previously Presented) The method of Claim 12, wherein the built software product comprises execution code that is generated from a source code component stored in the central repository.

17. (Previously Presented) The method of Claim 12, further comprising providing a configuration management subsystem that controls and manages different versions of the software components stored in the central repository.

EVIDENCE APPENDIX
Application Serial No. 09/943,563
Filed August 30, 2001

No evidence is being submitted with this Appeal Brief pursuant to 37 C.F.R. §§ 1.130, 1.131 or 1.132.

RELATED PROCEEDINGS APPENDIX
Application Serial No. 09/943,563
Filed August 30, 2001

Appellant is aware of no interferences or appeals that would be affected by the present appeal.